

ePub^{WU} Institutional Repository

Rony G. Flatscher

Informationssystementwicklung mit CASE an der Wirtschaftsuniversität Wien
- Erfahrungen mit ORACLE*CASE

Other (Published)
(Refereed)

Original Citation:

Flatscher, Rony G.

(1994)

*Informationssystementwicklung mit CASE an der Wirtschaftsuniversität Wien - Erfahrungen mit ORACLE*CASE.*

Gabler Verlag, Wiesbaden.

This version is available at: <https://epub.wu.ac.at/8251/>

Available in ePub^{WU}: December 2021

ePub^{WU}, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.

This document is the publisher-created published version. It is a verbatim copy of the publisher version.



Informationssystementwicklung mit CASE an der Wirtschafts- universität Wien

Erfahrungen mit ORACLE*CASE

Von Rony G. Flatscher

Überblick

- Der Autor berichtet über die Verwendung von ORACLE*CASE (Computer-Aided Software Engineering) in der Ausbildung von Studenten zur Datenverarbeitungsfachkraft.
- Methodisch wird auf Erfahrungen aus einer vierstündigen Arbeitsgemeinschaft zurückgegriffen. Durch eine Befragung wurden subjektive Eindrücke der Studenten von der Arbeit mit CASE gewonnen.
- Für die Praxis ist der Umgang von Studenten mit Software-Tools interessant, weil in der Praxis intensiv geschulte und erfahrene Benutzer, nicht aber Anfänger eingesetzt werden, bei Anfängern aber die Probleme dieser Tools viel klarer zutage treten.
- Für die Wissenschaft ist die Erkenntnis wichtig, daß noch so gute und fundierte theoretische Veranstaltungen den praktischen Umgang mit den Software-Tools an einem einfachen Beispiel (Entwicklung eines Informationssystems für einen Sportartikel-Großhändler) ersetzen können.

Dr. Rony G. Flatscher, Wirtschaftsuniversität Wien, Institut für Informationsverarbeitung und Informationswirtschaft, Abteilung für Wirtschaftsinformatik, Augasse 2-6, A-1090 Wien.

ZfB
ZEITSCHRIFT FÜR
BETRIEBSWIRTSCHAFT

© Gabler-Verlag 1994

A. Übersicht über das Lehrprogramm der Abteilung für Wirtschaftsinformatik

Im Rahmen des Studiums im zweiten Studienabschnitt bieten die Abteilungen für Wirtschaftsinformatik die Spezielle Betriebswirtschaftslehre¹ *Wirtschaftsinformatik* im Ausmaß von 12 Wochenstunden an.

Die Abteilung für Wirtschaftsinformatik ist Bestandteil des Instituts für Informationsverarbeitung und Informationswirtschaft an der Wirtschaftsuniversität Wien, das sich aus folgenden Abteilungen zusammensetzt:

- Abteilung für Wirtschaftsinformatik (Vorstand: Prof. H. R. Hansen),
- Abteilung für Angewandte Informatik insbesondere Betriebsinformatik (Vorstand: Prof. W. Janko),
- Abteilung für Industrielle Informationswirtschaft (Vorstand: Prof. A. Taudes).

Die Abteilung für Wirtschaftsinformatik bietet im Rahmen ihrer Speziellen Betriebswirtschaftslehre *Wirtschaftsinformatik* zwei unterschiedliche Schwerpunkte an, die es den Studenten ermöglichen, sich entweder auf das Tätigkeitsfeld *Datenverarbeitungsfachkraft* bzw. *EDV-Endbenutzer* in ihrem Studium zu konzentrieren. Abbildung 1 stellt die Gliederung des aktuellen Lehrprogramms der Abteilung dar.

Wir sehen das Tätigkeitsfeld *Datenverarbeitungsfachkraft* im Bereich des Managements der Informationssystementwicklung und unterrichten seit dem Wintersemester 1992/93 mit Hilfe des integrierten CASE-Systems ORACLE*CASE.

Das Tätigkeitsfeld *EDV-Endbenutzer* beschäftigt sich mit der Unterstützung von Endbenutzern auf allen Ebenen einer Organisation und setzt sich schwerpunktmäßig mit Endbenutzerwerkzeugen und deren Einsatz in Netzwerken auseinander. Das Lehrveranstaltungsprogramm für diesen Zweig wurde im Wintersemester 1993/94 aufgenommen.

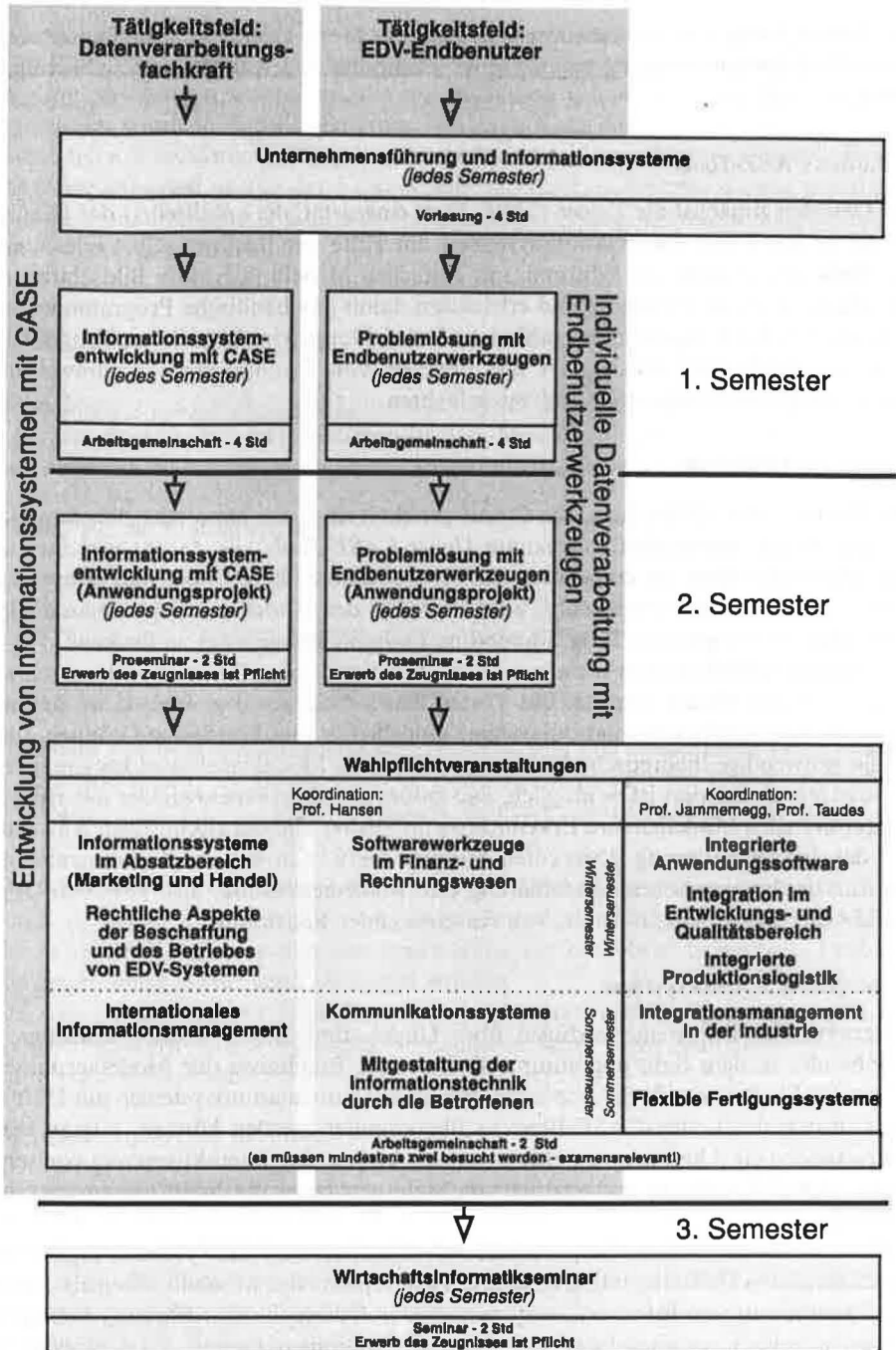
B. Informationssystementwicklung mit CASE

Im Rahmen der Speziellen Betriebswirtschaftslehre wurde an unserer Abteilung in der Vergangenheit sehr viel Wert auf die Erstellung von Informationssystemen mit Hilfe von Programmiersprachen gelegt. So wurde beispielsweise durch ein Jahrzehnt hindurch die für den kommerziellen Sektor klassische Programmiersprache COBOL (bis Wintersemester 1992/93) unterrichtet. Bereits in diesem Rahmen wurden Schwerpunkte im Bereich des Projektmanagements, der Softwarewartung und -dokumentation sowie der Einführung und Wartung von Softwaresystemen gesetzt.

Durch die laufende Beschäftigung mit den Entwicklungen im akademischen und im kommerziellen Bereich – im Zusammenhang mit der Erstellung von Informationssystemen – wurden in das Lehrprogramm der Abteilung laufend Erweiterungen eingebracht. So wurden u. a. folgende Modellierungstechniken in den Bereich der Informationssystementwicklung eingeführt:

- Entity-Relationship-Modellierung,
- Funktionsmodellierung inklusive Funktionshierarchien, Entscheidungstabellen, Pseudocode und Datenflußdiagramme.

Abb. 1: Gliederung der Speziellen Betriebswirtschaftslehre Wirtschaftsinformatik.



I. Computer Aided Software Engineering (CASE)

Die Entwicklung von Informationssystemen wurde im Laufe der Zeit immer stärker durch Rechnerunterstützung ergänzt (engl. Computer Aided Software Engineering, abgekürzt: CASE).

1. Lower-CASE-Tools

Hier wurden zunächst die *Lower-CASE-Tools* eingesetzt, die unmittelbar das händische Programmieren von Informationssystemen mit Hilfe der Rechner selbst erleichterten. Beispielsweise erlauben es Editoren, mit einfachen Mitteln interaktiv Bildschirm- oder Druckformulare zu entwerfen und erleichtern damit das händische Programmieren beträchtlich. In der Folge konnten zahlreiche Entwicklungen im Bereich von Programmgeneratoren beobachtet werden, die das Erstellen von Standardprogrammanweisungen immer weiter automatisierten und vereinfachten.

2. Upper-CASE-Tools

Parallel zu diesen Hilfsmitteln, die darauf gerichtet sind, das händische Programmieren zu minimieren, entstanden sogenannte *Upper-CASE-Tools*, die darauf gerichtet sind, Informationssysteme auf konzeptioneller Ebene zu modellieren, *ohne* daß es notwendig – oder theoretisch gar erwünscht – wäre, während der Modellierungsphase an die Notwendigkeiten der späteren Programmierung (= Implementierung) zu denken.

In diesem Bereich fanden vor allem Entity-Relationship- und Funktionshierarchiedialogmeditoren breiten Einsatz. Ein Vorteil dieses Ansatzes liegt u. a. darin, daß auch Nichtprogrammierer Informationssysteme modellieren bzw. konzipieren können, sofern sie das notwendige theoretische Rüstzeug besitzen, die Modellierungswerkzeuge sinnvoll einzusetzen. Zumindest ist es möglich, daß Informationssystementwickler mit Hilfe von konzeptionellen Modellen ihre Erkenntnisse im Dialog mit den zukünftigen Anwendern vor der Implementierung überprüfen können. Gerade in diesem Zusammenhang ist die Entkopplung zwischen Modellierung und Implementierung, also zwischen Upper-CASE- und Lower-CASE-Tools, von entscheidender Bedeutung.

3. Integrierte CASE-Systeme

Integrierte CASE-Systeme verfügen über Upper- und Lower-CASE-Werkzeuge, die aufeinander in dem Sinn abgestimmt sind, als die Ergebnisse der Modellierungsphase im Upper-CASE-Bereich für die Implementierung von Informationssystemen mit Hilfe von Generatoren des Lower-CASE-Bereichs übernommen werden können. Hierzu zählen beispielsweise die Ableitung von Datenbankstrukturen, von interaktiven und von stapelorientierten Programmen und letztlich von Menüstrukturen für die Informationssystembenutzer.

Ein wesentlicher Nutzen beim Einsatz von integrierten CASE-Systemen liegt in einer standardisierten Dokumentation sowohl der konzeptionellen Modellbildung als auch der Implementierung von Informationssystemen. Durch diese Standardisierung werden u. a. die Softwarewartung vereinfacht und das enge Zusammenarbeiten von zahlreichen In-

formationssystementwicklern im Team gefördert, da die einzelnen Mitarbeiter die Arbeitsergebnisse von anderen leicht – aufgrund der vom CASE-System vorgegebenen Standards – nachvollziehen können.

Dies führt u. a. zu neuen Märkten für sogenannte *Templates*, die betriebswirtschaftliche Problemlösungen beinhalten, die mit Hilfe von CASE-Systemen modelliert und dokumentiert wurden². Ein Käufer eines Templates kann das entsprechende konzeptionelle Modell mit verhältnismäßig geringem Aufwand an seine unternehmensspezifischen Bedürfnisse anpassen und in der Folge eigene Informationssysteme daraus generieren. Durch die standardisierte Dokumentation von Entwicklungsinformationen wird dieser Anpassungsvorgang erleichtert.

II. CASE in der Lehre

Der Einführung von CASE in der Lehre sind zahlreiche Erfahrungen im Zusammenhang mit Beratungstätigkeiten der wissenschaftlichen Mitarbeiter und der intensiven wissenschaftlichen Beschäftigung mit diesem Bereich in Theorie und Praxis vorausgegangen.

1. Gründe für die Einführung

Die Gründe, die zur Einführung von CASE in der Lehre zu diesem relativ frühen Zeitpunkt geführt haben, sind mannigfaltig. In den folgenden beiden Abschnitten werden die für uns zwei wichtigsten Hypothesen kurz diskutiert.

a) Fehlendes Knowhow in der Wirtschaft

Im Rahmen von Beratungsprojekten und Diskussionen mit Teilnehmern an verschiedenen Veranstaltungen der österreichischen Arbeitsgemeinschaft für Datenverarbeitung, e. V. wurden folgende Problemfelder definiert, die den Einsatz von CASE-Systemen in der Praxis einschränken bzw. behindern:

- Der Berufseintritt der derzeit aktiven Informationssystemmanager erfolgte in der Regel zu einem Zeitpunkt, zu dem CASE-Knowhow bzw. die dafür notwendigen theoretischen Kenntnisse und ihre Anwendung auf betriebswirtschaftliche Problemstellungen nicht ausreichend vermittelt wurden.
- Die Arbeitsbelastung von Praktikern führt häufig dazu, daß die Weiterbildung für das Erlernen zeitaufwendiger Konzepte vernachlässigt wird, wenn nicht unmittelbar wirksame Nutzeffekte damit verbunden werden können.

b) Wettbewerbsvorteile für unsere Absolventen

Sofern die Hypothese zutrifft, daß in der Wirtschaft CASE-Knowhow fehlt, sollte dies gleichzeitig ein Wettbewerbsvorteil für unsere Absolventen darstellen, die aufgrund der Lehrplananordnung frühestens im Herbst 1994 in den Arbeitsmarkt eintreten werden. Wir erwarten, daß Organisationen im Zeitverlauf verstärkt integrierte CASE-Systeme für die Eigenentwicklung von Informationssystemen einsetzen und daher verstärkt entsprechend ausgebildete Studenten nachfragen.

In einer Situation, in der die Ausbildung in diesem Bereich auf Hochschulebene aus verschiedenen Gründen noch vereinzelt erfolgt, erhöht eine praxisnahe Ausbildung in der Handhabung von integrierten CASE-Systemen den Wettbewerbsvorteil für unsere Studenten am Arbeitsmarkt.

2. Vorbereitungsarbeiten

a) Auswahl von ORACLE*CASE

Anfang 1992 wurde die Auswahl für das CASE-System der Firma ORACLE getroffen. Es handelt sich dabei um ein integriertes CASE-System eines etablierten und erfolgreichen Datenbankherstellers, so daß eine laufende Weiterentwicklung gewährleistet erscheint. Zum anderen entstehen durch die Sponsortätigkeit der Firma ORACLE der Abteilung für Wirtschaftsinformatik für einen Zeitraum von 10 Jahren keinerlei Softwarekosten in bezug auf Upgrades oder bei einem Wechsel der eingesetzten Betriebssystem- oder Hardwareplattformen. Darüber hinaus hatten Mitarbeiter der Abteilung im Rahmen des WUFIS-Projekts³ sowohl Anwendungs-Knowhow mit der Datenbank als auch mit den 4. Generationssprachen von ORACLE erworben⁴.

b) Ausbildung der Vortragenden

Vier Mitarbeiter der Abteilung für Wirtschaftsinformatik besuchten über Wochen hinweg die entsprechenden Kurse der Firma ORACLE, um das CASE-System kennen- und handhaben zu lernen. Es wurden die beigegebenen Lehrunterlagen kritisch analysiert und die Kursteilnehmer anderer Firmen über ihre Erfahrungen interviewt.

Aufgrund dieser Erfahrungen und unter Berücksichtigung didaktischer Aspekte wurde der inhaltliche Schwerpunkt für die geplante vierstündige Arbeitsgemeinschaft und die daran anschließende zweistündige Übung festgesetzt.

c) Entwurf eines Beispiels

Die Grundlage für die Arbeit in der Arbeitsgemeinschaft, die in die Handhabung und Anwendung des CASE-Systems einführt, bildet ein kleines, betriebswirtschaftlich orientiertes Beispiel für eine Auftragsverwaltung.

Aufgrund der Beispielangaben soll ein kleines Informationssystem für einen Sportartikelgroßhändler erstellt werden. Der Text erlaubt die Modellierung des Entity-Relationship-Modells mit Entitätstypen für Kunden, Vertreter, Artikel und Aufträge mit Bestellungen. Für die Modellierung der Funktionshierarchie sind zumindest einfache Wartungsfunktionen für alle Entitätstypen vorzusehen. Das Ziel liegt in der Entwicklung eines ablauffähigen Informationssystems, das sich aus den Teilsystemen für die Stammdatenverwaltung und die Auftragsverwaltung zusammensetzt.

3. Lehrveranstaltungsgliederung

Die vierstündige Arbeitsgemeinschaft vermittelt die Grundlagen der Handhabung eines CASE-Systems zur Entwicklung von betriebswirtschaftlichen Anwendungssystemen. Die Lehrveranstaltungsliederung dafür wird in dieser Arbeit kurz vorgestellt⁵.

Die im Folgesemester anschließenden zweistündigen Übungen haben lediglich das Ziel, die unterrichteten Konzepte zu vertiefen, indem von den Studenten für neue Problemstellungen Informationssysteme selbständig zu entwickeln sind.

Voraussetzung für den Besuch der CASE-Lehrveranstaltungen ist der Besuch der Grundlagenvorlesung *Unternehmensführung und Informationssysteme*, die sich an Österle et al. (1991) orientiert und zusätzlich u. a. die Grundlagen zur Daten- und Funktionsmodellierung beinhaltet.

- Einführung in die CASE-Terminologie: Es werden die grundlegenden Begriffe CASE, Upper-CASE, Lower-CASE und I-CASE erklärt.
- Einführung in das Vorgehensmodell: Es wird das Vorgehen erklärt, das für die Entwicklung von Informationssystemen prinzipiell gelten soll. Daran anschließend wird auf das Vorgehensmodell von ORACLE*CASE eingegangen, das die Phasen Strategie, Analyse, Design und Implementierung beinhaltet. Hierbei zählen die Phasen Strategie und Analyse zum Upper-CASE, Implementierung zum Lower-CASE. Die Designphase weist eine Brückenfunktion zwischen Upper- und Lower-CASE auf, indem sie zur Implementierung überleitet. In der Strategieweise werden Unternehmensziele, Geschäftseinheiten und kritische Erfolgsfaktoren definiert, die in weiterer Folge für die Entwicklung von Informationssystemen berücksichtigt werden müssen. In der Analysephase werden die Daten- und Funktionsanforderungen für den zu modellierenden Realitätsausschnitt der Organisation entwickelt. Die Designphase verfeinert die aufgefundenen Modelle soweit, daß in der Folge in der Implementierungsphase Datenbankstrukturen und Informationssysteme generiert werden können.
- Einführung in die Entity-Relationship-Modellierung: Die Grundlagen der Entity-Relationship-Modellierung (abgekürzt: ERM) nach Navathe/Elmasri werden wiederholt. Daran anschließend werden die Implementation der ERM von ORACLE*CASE erklärt und die Unterschiede zu Navathe/Elmasri herausgearbeitet. Kleine Übungen mit dem ER-Diagrammeditor beschließen diesen Schwerpunkt.
- Einführung in die Funktionsmodellierung: Die Grundlagen der Funktionsmodellierung für die Erstellung von Funktionshierarchien und von Datenflußdiagrammen werden wiederholt. Nach der Vorstellung der Implementation in ORACLE*CASE lernen die Studenten mit Hilfe kleinerer Übungen, die entsprechenden graphischen Editoren zu handhaben.
- Einführung in die Matrixtechnik von ORACLE*CASE: Die Studenten lernen die Möglichkeiten kennen, die graphische Editoren für die Verknüpfung von Daten, Funktionen, Geschäftseinheiten u. ä.m. einräumen. Dadurch stehen eine Vielzahl von weiteren Auswertungsmöglichkeiten unter unterschiedlichsten Gesichtspunkten zur Verfügung, mit deren Hilfe die Modellierungsergebnisse überprüft werden können.
- Erzeugen von DDL⁶-Scripts: Für die Generierung von DDL-Scripts aus Entity-Relationship-Modellen werden die Ableitungsregeln erläutert. DDL-Scripts stellen SQL⁷-Anweisungen für relationale Datenbanken dar, mit deren Hilfe Datenbankstrukturen definiert werden können, beispielsweise Tabellen, Indizes, Primärschlüssel, referentielle Integritätsbedingungen u. ä.m. Die Möglichkeiten, die ORACLE*CASE für die Beeinflussung dieser Ableitungen einräumt, werden erläutert und deren Implikationen für die Generierung von Informationssystemen diskutiert.

- CASE-Dictionary vs. Data-Dictionary: Die Unterschiede zwischen beiden Dictionaries werden erläutert, wobei wir aus Gründen der leichteren Unterscheidbarkeit für das CASE-Dictionary den Begriff Repository verwenden. Ein Data-Dictionary bezeichnet alle Strukturen, die in einer relationalen Datenbank enthalten sind und in denen alle Daten der Datenbank gespeichert sind. Im Unterschied dazu besteht ein CASE-Dictionary aus Datenstrukturen, in denen nur jene Daten enthalten sind, die im Zusammenhang mit CASE selbst relevant sind. So finden sich beispielsweise in einem Repository die Definitionen für ER-Modelle, die Datenbankstrukturen, die daraus abgeleitet wurden und in der Folge dazu verwendet werden, um im Data-Dictionary Tabellen für die Informationssysteme anzulegen⁸.
- Modulbildung und Generierung von SQL*Forms-Applikationen: SQL*Forms ist eine Programmiersprache der 4. Generation von ORACLE und stellt das Zielsystem für interaktive Informationssysteme unter ORACLE*CASE dar. Ausgangspunkt der Generatoren sind jene Module, die die Studenten definieren. In der Folge benutzen die Studenten die generierten Applikationen, um Daten zu erfassen und zu warten.
- Kurzeinführung in SQL*Plus: SQL*Plus stellt eine ORACLE-eigene Erweiterung zum Sprachstandard SQL dar, die ein interaktives Arbeiten mit der relationalen Datenbank erlaubt. Die Studenten sollen in die Lage versetzt werden, mit dem SQL-Sprachstandard einfache Abfragen durchzuführen, die ihnen ihre über die Applikationen eingegebenen Daten darstellen.
- Modulbildung und Generierung von Berichten (Reports): Mit Hilfe der Generatoren sollen die Studenten Berichte erzeugen, wobei neben SQL*Plus-Generatoren auch jene für den SQL*Reportwriter von ORACLE zum Einsatz kommen.

4. Lehrveranstaltungsziele

Der Schwerpunkt der Arbeitsgemeinschaft liegt in der Vermittlung von Grundlagenwissen für die Modellierung von Informationssystemen und auf praktischen Übungen, mit denen das erworbene Wissen auf eine kleine, betriebswirtschaftliche Problemstellung angewandt werden muß. Über 50 % der zur Verfügung stehenden Zeit wird dafür budgetiert.

Damit soll das erworbene Grundlagenwissen grundsätzlich auf CASE-Systeme anderer Hersteller übertragbar sein. Auf der anderen Seite soll durch die Konzentration auf die Analyse den Studenten bewußt gemacht werden, wie wichtig diese Phase für die erfolgreiche Modellierung von Informationssystemen in weiterer Folge ist.

Die restliche Zeit wird darauf verwendet, Module zu definieren, für die in der Folge die ORACLE-bezogenen Generatoren zum Erstellen von Anwendungen eingesetzt werden. Dieser zeitaufwendige Prozeß erscheint uns deshalb als sehr wichtig, damit die Studenten selbst *erfahren*, wie ihre Analysefähigkeiten und ihre Modellbildungen zu einsatzfähigen Informationssystemen transformiert werden. Eine wichtige Erkenntnis für die Studenten soll darin liegen, daß Schwächen in der Analysephase regelmäßig Inkonsistenzen und Probleme in späteren Phasen nach sich ziehen. Das Wissen um die ORACLE*Generatoren kann allerdings nicht direkt auf CASE-Systeme anderer Hersteller übertragen werden.

Die Entwicklung mit Hilfe von CASE-Systemen erfolgt in der Regel in Teamarbeit. In der gleichen Situation befinden sich die Studenten während der Lehrveranstaltungen, sodaß die soziale Dimension unmittelbar für sie erfahrbar wird.

5. Erfahrungen

Die erste vierstündige Arbeitsgemeinschaft wurde im Wintersemester 1992/93 durchgeführt, die ersten zweistündigen Übungen im Sommersemester 1993.

a) Lehrveranstaltungsleiter

Die Herausforderung für die vier Lehrveranstaltungsleiter⁹ der Arbeitsgemeinschaft war groß. Ein mächtiges, integriertes CASE-System bedeutet, daß es über eine Unzahl an Funktionen verfügt, die nur mit großem zeitlichem Aufwand erlernt werden können.

ORACLE*CASE besteht aus etwa 120 unterschiedlichen Funktionen, die selbst zum Großteil eigenständige Applikationen darstellen. Vom Hauptmenü ausgehend, sind alle Funktionen/Applikationen in 620 (!) Menüpunkten¹⁰ organisiert. Zudem existieren für die Generatoren mehr als 100 Parameter, die das Verhalten der einzelnen ORACLE*CASE-Werkzeuge mehr oder weniger stark beeinflussen. Denselben Anforderungen stehen auch die Studenten gegenüber.

Aufgrund der Komplexität hat es sich bewährt, daß mehr als ein Lehrveranstaltungsleiter die Arbeitsgemeinschaft betreut hat. Im ersten Semester waren es vier, in der Folge zwei Lehrveranstaltungsleiter. Dadurch ist es möglich, die Studenten andauernd zu motivieren, nachdem mindestens ein weiterer Leiter direkt auf die einzelnen Studenten eingehen kann. Diese Motivierung erscheint uns für den Erfolg der Lehrveranstaltung unumgänglich.

b) Studenten

Die Komplexität eines integrierten CASE-Systems führte zu einer Reihe von Verständnisproblemen.

So waren den Studenten das Zusammenspiel und die Integration der verschiedenen Phasen bei der Entwicklung von Informationssystemen nicht klar, obwohl das Vorgehensmodell theoretisch ausführlich erklärt wurde. Das Verständnis für die Zusammenhänge zwischen der Strategie-, Analyse-, Design- und Implementierungsphase, sowie die Wichtigkeit – vor allem der Analysephase – wurde den Studenten erst durch ihre eigenen, praktischen Erfahrungen mit dem CASE-System klar. Diese Erkenntnisse waren den Studenten erst gegen Ende des Semesters möglich, als die Implementierung mit Hilfe der Generatoren zu einem internalisierten Problembewußtsein über die Phasen führte, die sie erst zu diesem späten Zeitpunkt alle durchlaufen hatten. So konnten Schwächen bei der Implementierung von den Studenten selbst auf Schwächen in ihrer eigenen Analysephase zurückgeführt werden.

Die Menüstruktur von ORACLE*CASE ist auf intensive und erfahrene Benutzer ausgelegt, die Anfängern unübersichtlich erscheinen muß. Beispielsweise finden sich eine Reihe unterschiedlicher Menüpunkte doppelt und mehrfach in verschiedenen Ästen des Menübaums. Außerdem ist es möglich, daß im Menübaum einfach in einen anderen Ast

verzweigt wird, in dem eigentlich die Menüpunkte anderer Phasen enthalten sind. Gerade am Anfang führt diese Unübersichtlichkeit zu unbemerkten Phasenwechseln, die die Studenten verwirren.

Am Ende des Semesters wurden die Studenten über ihre Eindrücke befragt. Sie empfanden die Lehrveranstaltung als sehr aufwendig, bedingt durch die hohe Komplexität von ORACLE*CASE, die durch den vorhandenen Funktionsreichtum gegeben ist. Der Mehrzahl der Studenten wurden die Zusammenhänge zwischen den verschiedenen Phasen erst gegen Ende des Semesters bewußt, als sie wiederholt kleinere, isolierte Applikationen generierten. Als besonders schwierig erschien ihnen die *Anwendung* der erlernten theoretischen Modellierungskonzepte auf *betriebswirtschaftliche Problemstellungen*.

Trotz des großen Aufwandes für die Studenten ist die Dropout-Rate mit etwa 25 % relativ gering. Beispielsweise beendeten von 32 Studenten im Wintersemester 1992/93 24 Studenten die Lehrveranstaltung. Ein kleiner, informeller Fragebogen am Ende des Wintersemesters 1992/93, den 19 Studenten anonym ausfüllten, ergab folgende Einstellungen zu CASE-Systemen:

- 15 Studenten (79 %) halten integrierte CASE-Systeme für einen zukunftsweisenden Ansatz, Informationssysteme zu entwickeln,
- 14 (74 %) halten CASE-Systeme für den Einsatz in der Praxis geeignet,
- 13 (68 %) finden, daß sehr schnelle Entwicklungszeiten mit CASE-Systemen möglich sind,
- 12 (63 %) halten die standardisierte Dokumentation von Informationssystemen mit Hilfe von CASE-Systemen für wichtig,
- 15 (79 %) finden, daß der Ausbildungsstand der Praktiker und 14 (74 %) finden, daß der hohe Lernaufwand den breiten Einsatz von CASE-Systemen in der Praxis behindert.

14 (74 %) der Studenten hatten vor dem Besuch der Arbeitsgemeinschaft bereits praktische Erfahrungen mit der Entwicklung von Informationssystemen in unterschiedlichen 3. Generationssprachen.

Einige weitere bemerkenswerte Aussagen der Studenten über die Lehrveranstaltung und über CASE-Systeme im allgemeinen waren: „Schade, daß man das Programmieren dadurch nicht lernt!“, „... ein klassischer Programmierer ist mit CASE sicher überfordert!“, „CASE ist ein sehr gutes und effizientes Werkzeug ..., nur muß man vorher CASE beherrschen lernen, sonst fällt man mit CASE auf die Schnauze.“, „Vielleicht kann man damit mal verdammt viel Geld verdienen.“.

c) Infrastruktur

Die Lehrveranstaltungen zu CASE finden im Rechnerraum des Instituts für Informationsverarbeitung und Informationswirtschaft statt. Dieser Rechnerraum besteht derzeit aus 30 DEC-Ultrix-Maschinen, wovon drei Rechner als Server fungieren. Die Vorbereitung der Lehrveranstaltungsleiter erfolgte zunächst ausschließlich auf DEC-Ultrix-Maschinen und später teilweise auf normalen PC unter OS/2.

Die ursprüngliche Konfiguration zu Beginn des Wintersemesters 1992/93 war in der Ausstattung der Server zu wenig leistungsfähig, was zu langen Wartezeiten bei der

Nutzung von ORACLE*CASE führte. Letztendlich wurde die Konfiguration des Rechnerraums ausgebaut und stellt sich nunmehr wie folgt dar:

- 2 Server DECstation 5000/150 (R4000/50MHz) mit 32 MB RAM, CD-ROM- und Floppylaufwerk,
- 1 Datenbankserver DECstation 5000/150 (R4000/50MHz) mit 80 MB RAM, CD-ROM- und Floppylaufwerk,
- 27 Dataless DECstation 5000/120 (R3000/20MHz) mit 12 MB RAM und einer 120 MB Festplatte für das lokale Swappen.

Alle DEC-Ultrix-Maschinen verfügen über 19 Zoll Schwarz/Weiß Bildschirme und sind untereinander mit TCP/IP über ein Thin-LAN-Ethernet vernetzt.

Die Verwaltung von ORACLE*CASE für den Lehrbetrieb ist sehr aufwendig, nachdem für die Arbeit von vielen CASE-Benutzern die entsprechenden Scripts fehlen. Hinzu kommt, daß die Übersicht über die Datenbankimplementierung von ORACLE*CASE mangelhaft ist, so daß zeitaufwendige Analysen der implementierten Tabellenstrukturen und die Erstellung von entsprechenden Verwaltungsscripts notwendig sind. Nicht alle Verwaltungstätigkeiten können automatisiert werden. Beispielsweise müssen die entsprechenden CASE-Zugriffsberechtigungen an die Studenten manuell vergeben werden. Hinzu kommt, daß die Wartung der angelegten Datenbankobjekte aufwendig ist, nachdem in der eingesetzten Version der relationalen Datenbank eingetragene Benutzer eigentlich nicht gelöscht werden können. Die Defaultparameter für die Konfiguration der relationalen Datenbank sind in keiner Weise ausreichend, um ORACLE*CASE mit 20 oder mehr Personen zu unterrichten, da extrem hohe Wartezeiten für die Studenten entstehen. In der Folge mußten Studenten bis zu 20 Minuten warten, ehe die Ergebnisse von Generatorläufen verfügbar wurden. Die Kombination der oben besprochenen Erweiterung der Hardware und das Erhöhen von Datenbankparameterwerten führten dazu, daß im schlechtesten Fall die Wartezeit nur mehr fünf Minuten beträgt.

Folgende Softwareprodukte der Firma ORACLE werden derzeit eingesetzt:

- ORACLE RDBMS V6.0.34.2.1,
- SQL*Plus: Version 3.0.11.1.1,
- PL/SQL Version, 01.00.34.00.01,
- Oracle Toolkit Version, 01.00.19.00.03,
- SQL*Menu Version, 05.00.11.08.01,
- SQL*ReportWriter, Version 1.1.12.0.2,
- CASE*Dictionary, Version 5.0.22.1.1.

6. Schlußfolgerungen

Aufgrund der gesammelten Erfahrungen lassen sich einige Schlußfolgerungen ziehen, die anderen Lehrstühlen möglicherweise Anhaltspunkte geben können.

a) Lehrveranstaltungsleiter

Aufgrund der hohen Komplexität der zu beherrschenden Materie erscheint es sinnvoll, mindestens zwei Lehrveranstaltungsleiter für die Abhaltung derartiger Lehrveranstaltungen

gen vorzusehen. In unserem Fall können die Lehrveranstaltungen nur erfolgreich durchgeführt werden, wenn die Leiter gleichzeitig Experten auf folgenden Gebieten sind:

- Verwaltung (Aufsetzen, Parametrisierung, Wartung, physisches Datenbanklayout) der relationalen Datenbank,
- Modellierung von Daten und Funktionen (Entity-Relationship, Funktionshierarchien, Datenflußdiagramme),
- ORACLE-Generatoren und die dafür benutzten 4. Generationssprachen von ORACLE (SQL*Forms, SQL*Menu, SQL, SQL*Plus, SQL*Reportwriter).

Wenn mehrere Personen derartige Lehrveranstaltungen vorbereiten und leiten, können diese die unterschiedlichen Gebiete, für die detailliertes Expertenwissen notwendig ist, untereinander aufteilen. Bei Problemen ergibt sich durch diese Arbeitsteilung auch die Möglichkeit, auf Kollegenebene Hilfe von den anderen anfordern zu können.

Alle Erkenntnisse im Zusammenhang mit der Datenbank und dem CASE-System müssen sofort prägnant und übersichtlich dokumentiert werden, damit diese für alle Lehrveranstaltungsleiter zu einem späteren Zeitpunkt zur Verfügung stehen.

Sofern eine derartige Lehrveranstaltung zum ersten Mal vorbereitet wird, ist eine Vorbereitungszeit von mindestens drei bis sechs Monaten vorzusehen, damit in der Folge erfolgreich unterrichtet werden kann.

In jedem Fall müssen die Lehrveranstaltungsleiter im besonderen Maße darauf Wert legen, die Studenten andauernd positiv zu motivieren, um Frustrationserlebnisse auszugleichen.

b) Studenten

Studenten, die eine derartige CASE-Lehrveranstaltung zum ersten Mal besuchen, müssen von Haus aus ein besonderes Interesse daran haben, damit sie über die Zeitdauer der Lehrveranstaltung hinweg erfolgreich motiviert werden können. Desgleichen müssen sie ein hohes Maß an Einsatzbereitschaft mitbringen und über theoretische Vorkenntnisse zum Modellieren von Daten und Funktionen verfügen.

Die persönlichen Erfahrungen, die Studenten durch die praktische Arbeit mit CASE-Systemen vermittelt erhalten, können in keiner Weise durch theoretische Vorträge ersetzt werden. Nur durch abwechslungsreiche Kombination von theoretischem Vortrag *und* praktischer Übung am Computer kann jenes Problembewußtsein geschaffen werden, mit dem sie in späterer Folge erfolgreich Informationssystementwicklungen durchführen und leiten können.

c) Infrastruktur

Eine am modernsten Stand der Technik orientierte Infrastruktur erscheint unverzichtbar für einen erfolgreichen Unterricht mit CASE-Systemen. Die Hardware muß sich aus leistungsfähigen Servern, Arbeitsstationen und lokalem Netzwerk zusammensetzen.

Die Werkzeuge, die für die praktische Arbeit benötigt werden, sind im Falle von ORACLE*CASE sehr ressourcenintensiv, so daß sie wohl nur mit der adäquaten Infrastruktur erfolgversprechend einsetzbar sind.

Die derzeit bestehende Infrastruktur im Rechnerraum unseres Instituts, die weiter oben beschrieben ist, stellt vielleicht einen hilfreichen Ansatz dafür dar.

d) Lehrveranstaltungen

Für die Einführung und das Vermitteln des notwendigen Anwendungswissens erscheint uns eine *einführende vierstündige Arbeitsgemeinschaft* als absolute Minimalanforderung. Nach Möglichkeit empfiehlt sich sogar eine Ausweitung des Wochenstundenrahmens auf sechs bis acht Stunden. Auf keinen Fall ist es den Studenten zumutbar, daß sie sich das notwendige Wissen im Selbststudium aneignen. Für eine optimale Betreuung sind mindestens zwei Lehrveranstaltungsleiter zu empfehlen, so daß immer ein Leiter die Studenten an den Arbeitsstationen betreuen kann, während der andere Leiter Stoff vorträgt oder Arbeitsschritte vorführt. Die Studenten selbst sollten in Gruppen von zwei bis vier Personen zusammengefaßt werden, damit sie die Teamarbeit praktisch verwirklichen und einander helfen können, den umfangreichen Stoff zu verarbeiten und anzuwenden.

Die *Übungslehrveranstaltung* dient zur Vertiefung und Anwendung des in der Arbeitsgemeinschaft erarbeiteten Wissens und sollte einen Umfang von zwei bis vier Wochenstunden aufweisen. Um eine erfolgreiche Lehrveranstaltung zu gewährleisten, sollten in den Übungen nicht mehr als 12 Studenten unterrichtet werden, damit der Lehrveranstaltungsleiter alle Studenten intensiv auf persönlicher Basis betreuen kann. Auch für die Übungen sollten Teams von zwei bis vier Personen gebildet werden.

Anmerkungen

- 1 An der Wirtschaftsuniversität Wien können Studenten im 2. Studienabschnitt aus über 20 verschiedenen Speziellen Betriebswirtschaftslehren wählen.
- 2 Vgl. Rockart und Hofman (1992) über die steigende Bedeutung von Templates.
- 3 WUFIS stellt die Abkürzung für *WU-Führungsinformationssystem* dar und wird in der Dissertation von Pinzenöhler (1993) beschrieben.
- 4 Vgl. Flatscher (1993), S. 200 ff.
- 5 Vgl. Flatscher (1993b), S. 207 ff. über das detaillierte Lehrveranstaltungs-konzept.
- 6 DDL stellt die Abkürzung für *data definition language* dar und ermöglicht die Definition von Datenbankstrukturen.
- 7 SQL stellt die Abkürzung für *structured query language* dar und bezeichnet eine an die englische Sprache angelegte Abfrage-, Manipulations- und Definitionssprache für relationale Datenbanken.
- 8 In ORACLE*CASE bezeichnet CASE-Dictionary zudem die Applikation für ORACLE*CASE selbst.
- 9 Die Arbeitsgemeinschaft wurde von Dr. Brigitte Eichler, Dr. Rony G. Flatscher, Mag. Christian Marent und Dr. Andreas Pinzenöhler von der Abteilung für Wirtschaftsinformatik konzipiert und erstmals im Wintersemester 1992/93 gemeinsam abgehalten.
- 10 Einzelne Menüpunkte sind im Menübaum redundant. Oft erfolgt eine Verzweigung von einem Menüast in einen anderen.

Literatur

- Barker, R. (1989), CASE*METHOD: Entity-Relationship Modelling, New York.
 Barker, R. (1990), CASE*METHOD: Tasks and Deliverables, revised edition, New York.

- Barker, R. und Longman, C. (1992), CASE Method: Function and Process Modelling, New York.
- Batini, C., Ceri, S. und Navathe, S. B. (1992), Conceptual Database Design: An Entity-Relationship Approach, Redwood City.
- Elmasri, R. und Navathe, S. B. (1989), Fundamentals of Database Systems, Redwood City.
- Flatscher, R. G. (1993), Informationssystementwicklung mit CASE an der Wirtschaftsuniversität Wien am Beispiel von ORACLE*CASE – Motivation und Erfahrungen in der Lehre (Ausbildung), in: Tagungsband der ADV, Informations- und Kommunikationstechnologie für das neue Europa*, Wien.
- Flatscher, R. G. (1993b), The Curriculum for Teaching „Developing Information Systems with CASE“ at the Vienna University of Economics and Business Administration and the Experiences with Two Semesters of Teaching it, in: Proceedings of the International Association for Information Management 1993, Orlando.
- Jones, K. (1992), The lack of benefits realised through CASE, in: Tagungsunterlagen der ADV „CASE – Der Weg ist ein Ziel“, Wien.
- Olle, W. T., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. G., Van Assche, F. J. M. und Verrijn-Stuart, A. A. (1991), Information Systems Methodologies, 2. Auflage, Wolkingham.
- Österle, H., Brenner, W. und Hilbers, K. (1991), Unternehmensführung und Informationssystem – Der Ansatz des St. Galler Informationssystem-Managements, Stuttgart.
- Pinzenöhler, A. (1993), Innovative Informationssystementwicklung, gezeigt am Beispiel des WU-Führungsinformationssystems, Dissertation, Wirtschaftsuniversität Wien.
- Rockart, J. F. und Hofman, J. D. (1992), The Emerging Use of Application Templates, CISR Working Paper No. 250, Sloan Working Paper No. 3523-93, Sloan School of Management, Massachusetts Institute of Technology.

Zusammenfassung

An der Wirtschaftsuniversität Wien wurde im Wintersemester 1992/93 erstmals eine Lehrveranstaltung zur Informationssystementwicklung mit CASE von der Abteilung für Wirtschaftsinformatik für ihre Spezielle Betriebswirtschaftslehre *Wirtschaftsinformatik* im zweiten Studienabschnitt eingeführt. Diese Arbeit beschreibt in übersichtsmäßiger Form die Lehrveranstaltungsinhalte und die Erfahrungen, die gemacht wurden. Sie endet mit Schlußfolgerungen für die Lehrveranstaltungsleiter, zu den Studenten, zur Infrastruktur und zur Lehrveranstaltungsplanung selbst.

Summary

At the Vienna University of Economics and Business Administration the Department of Management Information Systems introduced courses for developing information systems with CASE in the fall semester of 1992/93. This article attempts to give an overview of the CASE-curriculum and the experiences, which have been gathered. It concludes with a short discussion of the requirements for the teachers, the students, the infrastructure and for the planning of such a course.

13: *Ausbildungs- und Berufsfragen (JEL J29)*

24: *Elektronische Datenverarbeitung (JEL M64)*